

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-148465
(P2000-148465A)

(43) 公開日 平成12年5月30日 (2000.5.30)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 9/06	5 4 0	G 0 6 F 9/06	5 4 0 F
9/445			4 2 0 M

審査請求 未請求 請求項の数1 OL (全7頁)

(21) 出願番号	特願平11-307087
(22) 出願日	平成11年10月28日 (1999.10.28)
(31) 優先権主張番号	09/182350
(32) 優先日	平成10年10月29日 (1998.10.29)
(33) 優先権主張国	米国 (US)

(71) 出願人	398038580 ヒューレット・パカード・カンパニー HEWLETT-PACKARD COMPANY アメリカ合衆国カリフォルニア州パロアル ト ハノーバー・ストリート・3000
(72) 発明者	ジョナサン・フィローズ アメリカ合衆国 コロラド州, ラプラン ド, パーク・コート 2702
(74) 代理人	100073874 弁理士 萩野 平 (外4名)

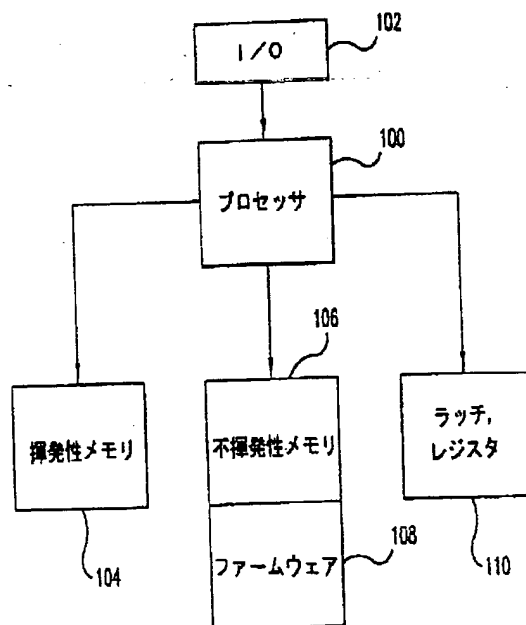
最終頁に続く

(54) 【発明の名称】 ファームウェア変更方法

(57) 【要約】

・【課題】 システムを再起動する必要なく、コンピュータシステムまたはオペレータに透明なファームウェアの更新をすること。

【解決手段】 ファームウェアの更新による影響を受けないメモリの異なる領域に、状態および構成情報を保存する。さらに、ファームウェア更新中に変化する可能性があり、不変を維持する必要がある情報は、ファームウェアの更新により影響を受けないメモリの異なる領域に記憶する。



【特許請求の範囲】

【請求項1】ファームウェアがメモリの第1のセクションに常駐しており、不変を維持する必要がある情報を、前記メモリの第1のセクション(108)からメモリの第2のセクション(104, 106)にコピーするコピーステップ(208, 308)と、前記メモリの第2のセクションを変更せずに前記メモリの第1のセクション内の前記ファームウェアを変更する変更ステップ(206, 310)と、を有することを特徴とするファームウェア変更方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般に、マイクロプロセッサ制御装置におけるファームウェア変更方法に係り、特に、装置の遮断を必要とせずかつシステム全体のリセットを必要としないファームウェア変更方法に関する。

【0002】

【従来の技術】マイクロプロセッサによって制御される多くの装置において、ファームウェアの修正または更新が必要になることがある。歴史的に、ファームウェアは、ROMに格納され、ファームウェアの修正は、少なくとも1つのROMを交換することによって行われていた。ROMの交換は、一般に、少なくとも製品の電源を落とすことを必要とし、製品を部分的に分解することが多い。最近では、様々なタイプのプログラマブル(追記型(write-once))、または再書き込み可能な不揮発性メモリが使用され、バッテリー式の揮発性メモリが使用されることもある。最新版のファームウェアを装置に電子的に送り、装置にそれ自体のファームウェアを修正または交換させることは現在一般的である。たとえば、モデムの最新版のファームウェアを電話回線を介して送信し、モデムのケースを開けることなくモデムのファームウェアを更新することができる。

【発明が解決しようとする課題】

【0003】一般に、稼働中のシステムのファームウェアを更新するとき、次のシステム・リセットまで変更されずに残さなければならない情報が消去又は変更されることがある。一般に、大きなシステム内の1つの装置のファームウェアを更新するとき、システム全体のリセット又は再起動を必要とすることがある。あるシステムは、1つの装置内のファームウェアを更新するとき、システム全体のリセットまたは再起動を必要とせずに一部のデータの保全性を維持し動作を継続する機能を必要とする。以下の説明では、ファームウェアの更新中にデータを保存する必要がある2つの装置の例を提供する。第1の例は、システム・リセット後の初期設定中に収集され送られるデータを含む。第2の例は、装置が他の装置との通信を必要とする可能性のあるデータ、又は外部操作に影響を与える可能性のあるデータを含む。

【0004】多くのコンピュータ・ベースのシステム(computer based system)は、電源投入後の初期設定プロセス中または再起動プロセス中に自動構成プロセスを経る。一般に、この自動構成プロセスは、すべての装置が同時にそれぞれ異なる初期設定プロセスを実行することを必要とする。たとえば、インテル互換のパーソナル・コンピュータ(Intel compatible personal computer)の場合、ISAバス用の入出力ボードを自動的に設定する業界仕様(industry specification)の1つは、プラグ・アンド・プレイISA規格と呼ばれている。ISAプラグ・アンド・プレイの場合、それぞれの互換入出力カード(compatible I/O card)が、ベンダ識別子及びシリアル番号を含む固有の識別子を有する。それぞれの互換入出力カードは、それ自体の識別子を読み取ることができる。ホストコンピュータは、最初、すべての互換入出力カードを構成モードにする。次に、ホストコンピュータは、それぞれの識別子内の連続したビット位置を示す一連の遷移(transition)により線(lines)を駆動する。ホストコンピュータと入出力カードとの間の対話の結果、1つの入出力カードが分離され、論理装置番号が割り当てられる。次に、このシーケンスは、すべての入出力カードに論理装置番号が割り当てられるまで繰り返され、別の入出力カードなどが分離される。

【0005】もう1つの一般的なインタフェース規格は、小型コンピュータ・システム・インタフェース(Small computer System Interface; 以下、SCSIと称す)である。SCSIもまた、各装置に固有のIDを必要とする。ある業界グループは、特に固有のSCSIのIDの自動割当てを提供するプラグ・アンド・プレイSCSIと呼ばれる一組の仕様を提案している。固有のIDを割り当てる特定の手順(protocol)は、自動構成SCSI(SCSI Configured AutoMagically; 以下、SCAMと称す)と呼ばれる。SCAM互換装置はそれぞれ、不揮発性記憶装置に記憶されたデフォルトIDを有する。SCAM主装置は、まず、他の各SCAM装置に1つつつコマンドを送り、非活動状態になる。次に、SCAM主装置は、ISAプラグ・アンド・プレイのプロトコルと類似のプロトコルを使用して、SCSIアドレスを割り当てるために各装置を分離する。

【0006】周辺機器には他の構成プロトコルも使用される。1つの入出力ポートに複数の装置がある場合、装置は、電源投入時に、自分自身を一次装置及び二次装置として自動的に設定することができる。装置は、自身の状態(一次または二次)と、別の装置が同じ入出力ケーブルを共用しているかどうかを記録しなければならない場合がある。

【0007】第2の例では、装置が他の装置と共有する情報又は様々な制御機能のために装置が使用するデータが、ファームウェアの更新のよって変更されることがある。電源投入時または再起動中にだけ情報が他の装置に

送信されると、ファームウェアの更新後に装置によって使用される情報は、一貫性がなかったり、不適切であることがある。この場合、システムの再起動が必要になることがある。例として、再び識別子を検討する。識別子の一部分は、ファームウェアのバージョンまたは日付コードを示すことがある。オペレーティング・システム・ソフトウェア又はその他のシステム装置は、電源投入時またはシステム再起動時に装置をポーリングし、装置識別番号のリストを記録することがある。アプリケーション・ソフトウェアは、このようなリストを読み取ることがあり、アプリケーション・ソフトウェアは、特定の装置識別子を後で参照することがある。ファームウェア最新版が、たとえばファームウェアのバージョン番号を変更することにより装置識別子を変更すると、システム再起動中にオペレーティング・システム又はその他のシステム装置によって収集された情報を更新するためにシステムを再起動が要求される。代替の例として、ファームウェアに記憶されておりリセット中に制御回路のレジスタに書き込まれる値を検討する。ファームウェアの更新中にこの値が変更され、ファームウェアが再起動されると、制御回路の出力が不適切に変化することがある。

【0008】本発明は上記事情に鑑みてなされたものであり、システムを再起動する必要がなく、コンピュータ・システムまたはオペレータにトランスペアレントなファームウェアの更新をすることを目的とする。

【0009】

【課題を解決するための手段】本発明によるファームウェア制御装置は、ファームウェアの更新による影響を受けないメモリの異なる領域に状態および構成情報を保存する。さらに、ファームウェアの更新中に変化する可能性があり、不変を維持する必要がある情報は、ファームウェアの更新により影響を受けないメモリの異なる領域に記憶する。第1の実施例では、メモリの異なる領域のデータは、ハードリセット（電源投入時のリセット（power-on reset）又はシステム全体のリセット）中に上書きされる。第2の実施例では、ファームウェアの更新の初めにメモリの異なる領域にデータのコピーが記憶される。

【0010】第1の実施例において、ファームウェア制御装置のリセット・プロセスは、2つの部分に分けられる。装置のリセット・プロセスの第1の部分において、メモリの別の異なる領域の内容が、ファームウェアから又は他の装置との対話によって、更新される。装置リセット・プロセスの第2の部分において、他のすべてのリセット機能が実行される。リセット・プロセスの第1の部分は、電源投入時のリセット中又はシステム全体のリセットにตอบสนองしてのみ実行される。特に、装置のリセット・プロセスの第1の部分は、ファームウェアの更新後には実行されない。

【0011】第2の実施例において、変更されるべきで

はないデータは、ファームウェア更新プロセスの一部としてメモリの異なる領域にデータのコピーが記憶される。次に、リセット・プロセスの一部として、装置は、ファームウェアの更新が行われたかどうかを確認する。ファームウェアの更新が行われた場合は、適切な宛先に、メモリの異なる領域の中のデータのコピーが記憶され、メモリの異なる領域の中のデータが消去される。

【0012】その結果、どちらの実施例の場合も、メモリの異なる領域の内容は、ファームウェアの更新中またはその後も乱されず（disturb）、装置のファームウェアの更新後にシステムを再起動する必要がない。

【0013】

【発明の実施の形態】図1は、本発明に係るファームウェアを更新可能なプロセッサシステムを示すブロック図である。プロセッサ100は、入出力（I/O）ポート102を介して外部（装置の他の部分又は外部の装置）と通信することができる。プロセッサ100は、揮発性メモリ104、不揮発性メモリ106及び他の装置回路内の様々なラッチ並びにレジスタ（110）に読み出し及び書き込みをすることができる。「揮発性メモリ」という用語は、記憶された情報が電源を落としたときに失われることを意味する。例えば、RAMであり、ラッチ及びレジスタも揮発性メモリである。「不揮発性メモリ」という用語は、記憶された情報が電源を落としたときに失われないことを意味する。例えば、読取り専用メモリ（ROM）、プログラム可能読取り専用メモリ（PROM）、消去可能読取り専用メモリ（EPROM）、コア・メモリなどの磁気装置を含み、たとえばテープやディスクなどの大容量記憶媒体を含むこともある。一般に、ファームウェア108は、不揮発性メモリ106に常駐する。しかしながら、後で説明するように、プロセッサ100は、揮発性メモリ内に常駐するコードを実行することもできる。ファームウェア108は、命令及びデータの両方を含んでもよい。新しいファームウェアまたはファームウェアの修正は、入出力ポート102で受け取ることができる。

【0014】第1の実施形態において、電源投入時の初期化シーケンス中又はシステム全体のリセットにตอบสนองしたりリセット中に、ファームウェア108として指定された領域以外のメモリの異なる領域（セクションに該当）に情報が保存される。さらに、情報は、ファームウェア108から読み出され、メモリの異なる領域に記憶されることがある。メモリの異なる領域は、揮発性メモリ104内又は不揮発性メモリ106内にあってもよい。ファームウェア108が更新されるとき、該メモリの異なる領域は、更新中又は更新後の装置のソフトリセット中にも影響を受けない。

【0015】第2の実施形態において、ファームウェアの更新中に、ファームウェアの更新中に情報が変更される場所から、メモリの異なる領域に重要な情報がコピー

される。ファームウェアの更新後、ファームウェアの更新が行われたかどうかを確認するために検査が行われる。ファームウェアの更新が行われた場合、メモリの異なる領域にコピーされた情報は、元の場所に書き込み(copy back)される。

【0016】以下の説明を容易にするために、例として自動構成装置を使用する。しかしながら、本発明は、システム全体の電源投入時のリセット又は再起動を必要とせず、ファームウェアを更新しなければならないすべての装置に適用可能である。例示の装置は、ファームウェアのバージョン番号を含む電子的に読み出し可能な識別を有する。例示の装置は、入出力ケーブルを少なくとも1つの他の装置と共有することができる。入出力ケーブルを共有する各装置は、同じ入出力ケーブル上に別の装置が存在するかどうか判断しなければならない。別の装置が存在する場合は、入出力ケーブルを共有する各装置は、どの装置が一次装置で、どの装置が二次装置かを判断するプロセスを実行しなければならない。この構成情報は、電源投入時の初期化中又はシステムの再起動中に判断及び保存されなければならない。

【0017】図2は、第1の実施形態に係るファームウェアを更新する方法を示す状態図及びフローチャートである。図2において、ファームウェアは、ハードリセット・プロセス(ステップ202)、ソフトリセット・プロセス(ステップ204)又はファームウェア更新プロセス(ステップ206)の3つのプロセスのうちの1つを実行することができるアイドル・プロセス(ステップ200)を有する。「ハードリセット」及び「ソフトリセット」という用語は、システム全体にも使用されることがあるが、この用語は、ファームウェア制御装置内のプロセスだけを指すこととする。前記アイドル・プロセス(ステップ200)は、プロセス分岐の単なる例である。プロセスが他のプロセスに分岐することができる方法はたくさんある。一般に、様々なプロセスへの分岐を制御するために、実行中のプロセスを中断すること(たとえば、リセット信号によって中断させる)又はプロセスが一定の条件若しくはフラグを調べることがある。

【0018】ハードリセットが必要な場合(たとえば、特定のコマンド、電源投入時の初期化又はシステム全体のリセットを指示するシステムの中断若しくはその他の信号)、ファームウェアは、内部装置固有のデータを、ファームウェアの更新中に変更を受ける不揮発性メモリから、ファームウェアの更新中に変更を受けないメモリの異なる領域にコピーを記憶する(ステップ208)。たとえば、装置の識別が自動構成中に使用され、装置の識別がファームウェアの更新中に変更されることがある。この例において、前記ステップ208で、装置の識別は、メモリのファームウェアの更新中に変更されないメモリの部分にコピーが記憶される。次に、自動構成中及び他のハードリセットまでの動作中、オリジナルでは

なくコピーが記憶されたバージョンが使用される。

【0019】メモリの異なる領域は、不揮発性メモリ104内にあってもよく、不揮発性メモリ106の一部でもよい。この選択は、不揮発性メモリに使用される技術に依存する。不揮発性メモリを部分的に修正できる場合は、ファームウェアは、データを、更新を受ける不揮発性メモリから更新を受けない不揮発性メモリの異なる領域にコピーを記憶することがある。しかしながら、不揮発性メモリ全体を書き換えなければならない場合は、データを揮発性メモリにコピーを記憶することができる。自動構成(ステップ210)後に、構成情報は、異なるメモリ(揮発性メモリ又はファームウェアの更新中に変更されない不揮発性メモリの部分)に保存される(ステップ212)。

【0020】ファームウェアを更新する信号又はコマンドを受け取ると、変更されない更新コードが、新しいファームウェア(たとえば、入出力ポートを介して)を読み取り、新しいファームウェアを不揮発性メモリ(ステップ206)に書き込む。不揮発性メモリ106を部分的に修正できる場合は、不揮発性メモリの一部分を、ファームウェアの更新中に変更されない更新コードのために予約することができる。次に、変更されない更新コードが、新しいコードを読み取り、古いコードと置き換わる。図1において、不揮発性メモリ106は、更新される異なるファームウェア・セクション108を有するように示されている。代替として、ファームウェアの更新コードが、不揮発性メモリから揮発性メモリにコピーが記憶され、次に、プロセッサ100が、ファームウェアの更新中に揮発性メモリ内の更新コードを実行してもよい。更新コードが揮発性メモリから実行される場合、不揮発性メモリ106全体が更新されることもある。システムによっては、不揮発性メモリが書き換え可能なものもあり、その場合、古いファームウェアが上書きされることもある。他のシステムでは、不揮発性メモリが追記形のこともあり、その場合、新しいファームウェアがメモリの新しい領域に書き込まれ、ポイントが、古いファームウェアではなく新しいファームウェアを指すように変更される。

【0021】ファームウェアが更新された後、ソフトリセット・プロセス(ステップ204)を実行する。ソフトリセット・プロセス(ステップ204)は、前記ステップ208及び前記212で保存された情報に影響を与えないリセット・プロセスを有する様々なタスクをすべて備える(ステップ)。たとえば、大容量記憶周辺装置の場合は、変換器を始点まで移動させなければならないことがある。

【0022】図2に示した方法は、すべての装置に適しているわけではない。たとえば、装置によっては、ハードリセット及びソフトリセットの等価物(equivalent)がないものもある。さらに、装置によっては、少し異なる

結果を必要とするものもある。たとえば、リセット・プロセス中に、ファームウェアが、制御回路によって使用される図1記載のレジスタ(110)に値を書き込むことを必要とする。ファームウェアの更新後、制御値は変更されてはならないが、リセット後は、現在の制御値を使用しなければならない。図3は、第2の実施形態に係るファームウェアを更新する方法を示す状態図及びフローチャートである。図3のファームウェア部分には、ソフトリセットがない。アイドル・プロセス300は、ハードリセット302又はファームウェア更新プロセス304に分岐することができる。ハードリセット後、ファームウェアは自動構成プロセス(ステップ306)を実行する。ファームウェアの更新中、情報がデフォルトの場所からメモリの異なる領域にコピーが記憶される(ステップ308)。たとえば、レジスタに書き込まれる値が、ファームウェアの場所からメモリの異なる領域にコピーが記憶される。そして、ファームウェアは更新され(ステップ310)、再起動される(ステップ312)。ハードリセットまたはファームウェアの更新の後、ファームウェアは、ファームウェアの更新が行われたかどうかを確認する(ステップ314)。たとえば、ファームウェアは、メモリの異なる領域を確認して、データを含むか消去されたかを確認する。ファームウェアの更新が行われていなかった場合、デフォルト値が使用される(ステップ316)。ファームウェアの更新が行われていた場合、ファームウェアは、前にメモリの異なる領域にコピーを記憶した値(すなわち、前記ステップ308でコピーが記憶された値)を使用する(ステップ318)。前記ステップ318の代わりに、値がレジスタに既に書き込まれている場合は、前記ステップ318で何も行わなくてよい。すなわち、前記ステップ316で宛先レジスタを更新する代わりに、前記ステップ318は、単にレジスタを乱さないようにしてもよい。図3に示したファームウェアにおいて、リセットが行われた場合は、新しい値が使用され、したがって、メモリの異なる領域に記憶された値は、適切な宛先(たとえばレジスタ)にコピーが記憶された後すぐに消去される(ステップ320)。

【0023】図2及び図3に示した2つの方法は、相互に排他的ではなく、組み合わせることができ、あるいは、それぞれの方法の一部分を適切に使用することができる。それぞれの図は、装置の必要性によってわずかに異なる解決策を示す。ある情報は、ソフトリセットを乗り切る(survive)ことを必要とし(図2)、他の情報は、リセット後にリフレッシュされるのではなくファームウェアの更新を乗り切ることを必要とすることがある(図3)。

【0024】要するに、前述の装置及び方法によって、異なるメモリに保存された情報は、ファームウェア更新プロセス及びファームウェアの更新直後の装置のリセッ

ト・プロセスによっても乱されない。その結果、システム全体の再起動プロセスが不要になる。ファームウェアから異なるメモリに転送された情報は、ソフトウェア更新後から次のシステムの再起動前まで不適切になっている可能性がある(たとえば、ファームウェアのバージョン番号が間違っている)が、この結果は、即時にシステムの再起動を必要とすることよりも好ましい場合がある。

【0025】本発明の以上の説明は、例示と説明のために提示された。これは、本発明を開示した厳密な形態を制限するものではなく、上記教示を参照して他の修正および変更を行うことができる。実施形態は、本発明の原理とその実際の応用例を最もよく説明し、それにより当業者が、熟慮された特定の利用法に適切な様々な実施形態と様々な修正において本発明を最もよく利用できるように選択し説明した。

【0026】以下に、本発明の実施の形態を要約する。
1. ファームウェアがメモリの第1のセクションに常駐しており、変更されてはならない情報を、前記メモリの第1のセクション(108)からメモリの第2のセクション(104, 106)にコピーするコピーステップ(208, 308)と、前記メモリの第2のセクションを変更せずに前記メモリの第1のセクション内の前記ファームウェアを変更する変更ステップ(206, 310)と、を有するファームウェア変更方法。

【0027】2. 初期設定プロセス中に情報を獲得する獲得ステップ(210)と、前記獲得ステップ(210)の情報を前記メモリの第2のセクションに保存するステップ(212)と、をさらに有する上記1記載のファームウェア変更方法。

【0028】3. メモリの第1のセクションのファームウェアを変更する方法であって、初期設定プロセス中に情報を獲得する獲得ステップ(210)と、前記獲得ステップ(210)からの前記情報をメモリの第2のセクションに保存する保存ステップ(212)と、前記メモリの第2のセクションを変更せずに前記メモリの第1のセクション内の前記ファームウェアを変更する変更ステップ(206)と、を有するファームウェア変更方法。

【0029】

【発明の効果】以上の説明から明らかなように、システムの再起動が必要なく、コンピュータシステムまたはオペレータにトランスペアレントなファームウェアの更新することができる。

【図面の簡単な説明】

【図1】本発明に係るファームウェアを更新可能なプロセッサシステムを示すブロック図である。

【図2】第1の実施形態に係るファームウェアを更新する方法を示す状態図及びフローチャートである。

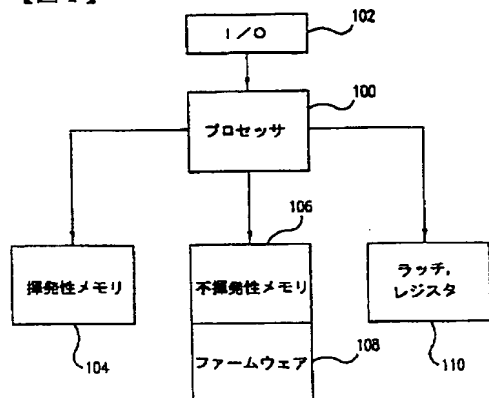
【図3】第2の実施形態に係るファームウェアを更新する方法を示す状態図及びフローチャートである。

【符号の説明】

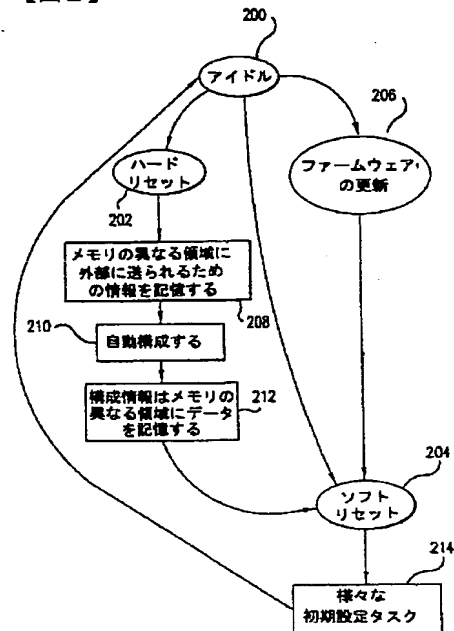
100 I/O
102 プロセッサ
104 揮発性メモリ
106 不揮発性メモリ

106 不揮発性メモリ
108 ファームウェア
110 ラッチ、レジスタ

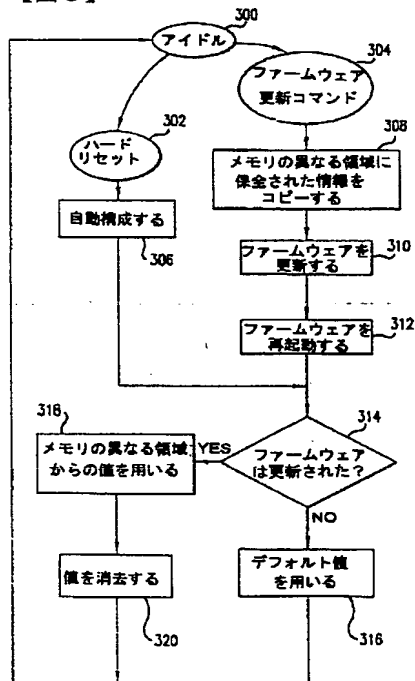
【図1】



【図2】



【図3】



フロントページの続き

(72)発明者 マイケル・オブライエン
アメリカ合衆国 コロラド州, ラブラン
ド・ダブリュー・18ティーエイチ・ストリ
ート 2306